A DATA COMMUNICATION INTEGRATED CIRCUIT

Here are details about the UART to help you in using it in a data terminal.

BY ROGER L. SMITH

PERHAPS you have already heard about the universal asynchronous receiver/transmitter—commonly referred to by its acronym UART. If not, you soon will because UART may become an important part of data-handling systems in all areas of electronics where information must be routed from one location to another. In a UART, the transmitter converts parallel data bits into serial form for transmission over two-wire lines. The receiver section does the reverse operation.

You could use a UART with a video display to communicate with a computer or another type of display. Coupling it with suitable circuitry, you could use a UART to record ASCII data from a keyboard on a tape recorder. Use it with a modem (modulator/demodulator) and telephone coupler or Data Access Arrangement (DAA), and you could transmit data over telephone lines. UART's can also be used in centrally monitored burglar and fire alarms, intersection traffic control, and ecological data gathering. In fact, the applications in which UART's can be used are almost limitless. (See Fig. 1 for diagrams of typical applications for the device.)

General Information. In Fig. 2 are shown the transmitting section of one UART (left) and the receiving section of another UART (right), in logic diagram format. For these UART's to operate properly together, they must both be referenced to the same clock rate, which must be 16 times the desired line transmission rate. For example, if we communicate with a Teletype®, the bit transmission rate is 110 bits/second. Therefore, the clock pulses delivered to pins 17 and 40 must be 110 × 16 = 1760 Hz. (The clocking signal can come from a crystal oscillator, but it is usually sufficient to use any stable oscillator that has an accuracy of 1% or better.

The UART has separate clock input pins for the receiver and transmitter sections so that receiving rates can be different from transmitting rates. For example, different rates might be used between terminals and from a terminal to a computer. Here, the transmission rate would be increased, to perhaps 1200 baud. (A baud is generally defined as a bit per second.) For another example, when the data comes from a manual keyboard, a rate of 150 baud should be adequate.

Before the UART can be operated, its internal registers and detectors must be cleared. This is usually accomplished automatically during power-up by pulsing reset pin 21 with a high (logic-1) pulse. You can do this with a resistor to ground and a capacitor to the +5-volt line.

Each half of the UART contains character-format mode control flip-flops that can be computer-controlled if the device is to be used with a computer. If the modes are to remain constant, the control pins can be hard-wired or connected to manual switches.

The controls that are available in a UART include:

I_{2SB} (pin 36)—transmitter stop bit control. A logic 0 causes one stop bit, and a logic 1 (+5 V) causes two stop bits to be transmitted.

I_{NP} (pin 35)—no-parity control. A logic 1 eliminates the parity bit from the transmitted data, disables the receiver parity check, and forces receiver parity error (0_{PE}) pin 13 to go to logic 0.

I_{PS} (pin 39)—parity select. A logic 0 inserts and checks odd parity, and a logic 1 inserts and checks even parity.

I_{NB1},I_{NB2} (pins 38, 37)—select character length of 5, 6, 7, or 8 bits/character.

I_{CS} (pin 34)—mode control strobe. A logic 1 enters the above controls into the holding register. This control can be hard-wired.

Once the control pins are either hard-wired or selected by computer, transmission can begin. Although the control pins serve both halves of the UART and both halves must operate in the same data format, they can be bussed in by the same lines that carry the data to be transmitted. The commands are then strobed into the holding flip-flops by pin 34; so, the computer can forget about them unless they must be changed.

Sending Data. Assuming all control pins have been selected, the transmission begins when a key is depressed. The ASCII, IBM Selectric®, Baudot, or other code appears at the UART's input pins (pins 26 through 33 if all eight bits are used). After a delay of 1 or 2 ms to allow the inputs to settle, a pulse must be sent to pin 23 (I_{DS}). This negative-going pulse is generated by

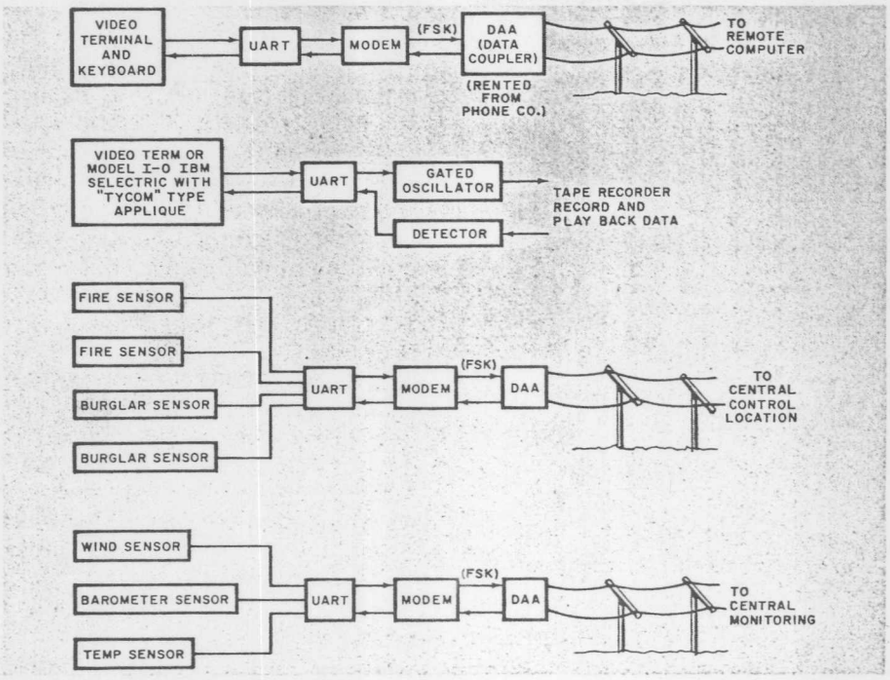


Fig. 1. Applications for UART include video systems, fire and burglar alarms, and weather monitoring.

logic on the keyboard. It enters the *data* into the input holding register. The transition of the pulse back to logic 1 causes a *start* bit to appear at serial output (0_{SO}) pin 25 after the next negative-going clock transition.

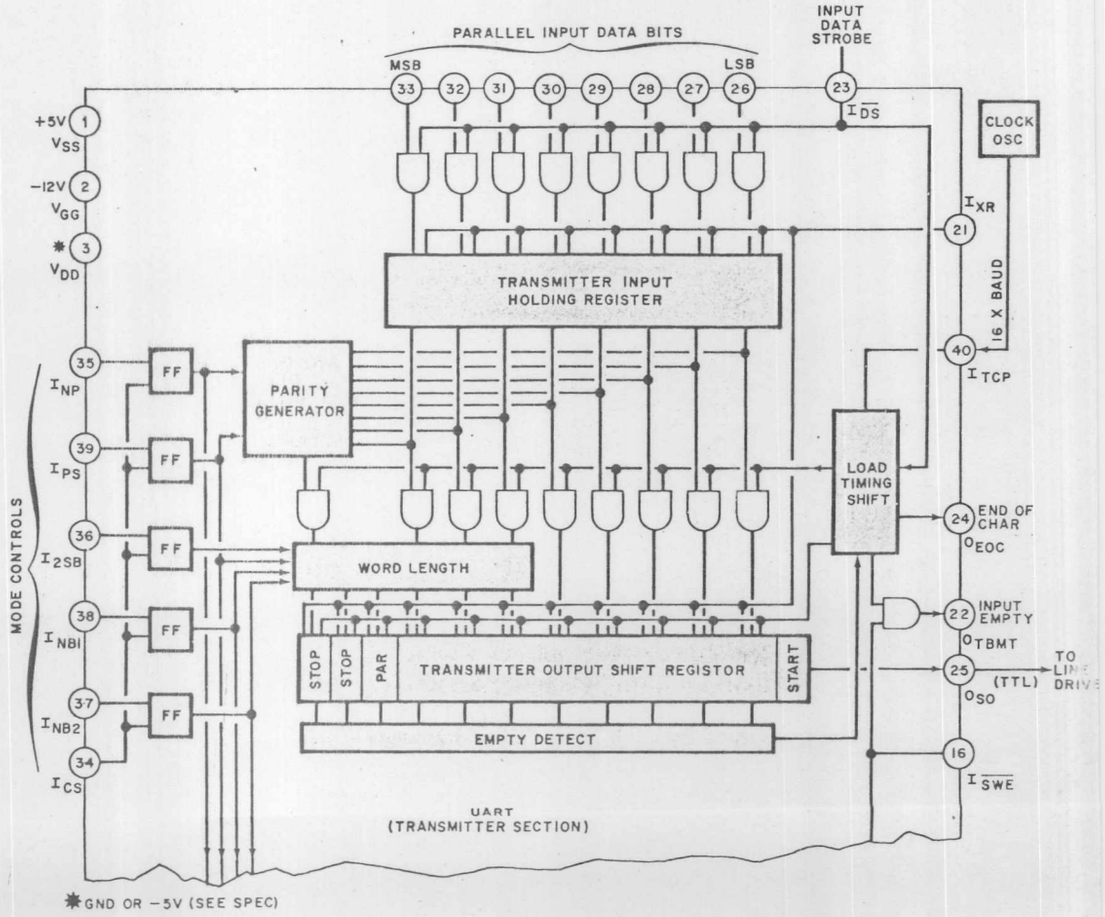
The UART is now transmitting the data out on pin 25 at a rate of one bit for every 16 clock pulses. The *start* bit is first, followed by the *data*, with the least-significant bit (LSB) first and the most-significant bit (MSB) last. This is followed by the *parity* and *stop* bits.

The UART executes the formatting commands as set up on the control pins. It then computes and adds the proper *parity* bit, if any. Just as it added a *start* bit preceding the *data*, it also adds one or two *stop* bits after the *data*. (The number of *stop* bits is determined by the logic level at pin 36.) If all possible bits were transmitted, they would number 12, including one *start*, eight *data*, one *parity*, and two *stop* bits. The minimum number transmitted can be seven: one *start*, five *data*, and one *stop* bits.

Since the transmitter section has an input holding register as well as an output shift register, it can receive a second keystroke input immediately following the first. If this should happen, both the holding and shift registers would be full and this would be indicated by a logic 0 on pin 22. This pin goes to logic 1 when the UART is ready to accept additional data (transmitter input holding register empty, 0_{TBMT}).

Normally, the output of the transmitter portion (pin 25) is boosted to the standard RS-232C interface levels to be sent to other devices. For example, the data could go to a modem to be

Fig. 2. At near right is the transmitting section of one UART. Receiving section of another UART is on opposite page.



converted to fsk (frequency-shift, or two-tone, keying). It can then go via either an acoustic coupler or a DAA to the phone line and some remote terminal or computer location where it is demodulated, converted to RS-232C and TTL levels, and then possibly to the receiver input of another UART.

Receiving Data. The TTL input level on pin 20 of the UART is normally logic 1, and the receiving section monitors this pin for any transition to logic 0. When a logic 0 transition occurs, a counter is started, clocked by the input on pin 17 at a 16 times baud rate.

When the counter reaches 8, which should be the center of the incoming bit, the UART again checks the level at pin 20 to ascertain that it is still at logic 0. If it is, a valid *start* bit has been received, and the UART begins to count in increments of 16 clock pulses to go from center to center of the incoming bit "cells." (Some designs check the input during all of the first 8 counts after a transition to logic 0. Whichever the case, noise spikes are not likely to cause false starts.)

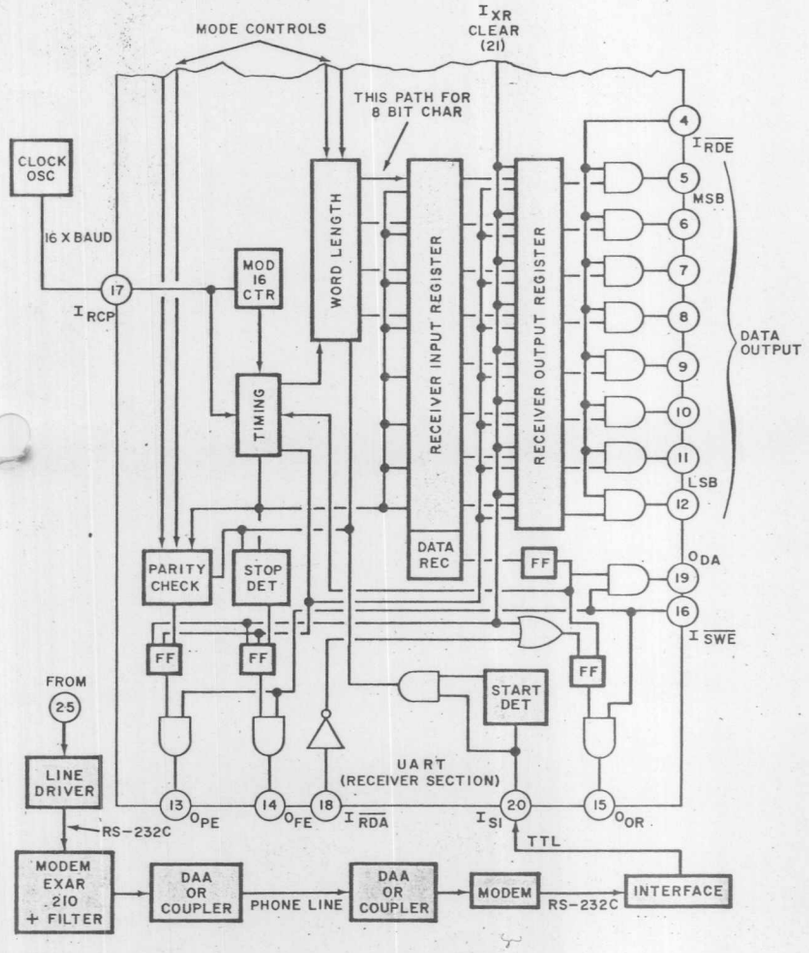
Each time the counter reaches 16, the center of a bit cell has been reached and a *shift* pulse is applied to the input register. In this manner, all *data* bits are loaded into the shift register (LSB first), followed by *parity* and *stop* bits. Since the character length command was previously set up on pins 37 and 38, internal gating insures that the incoming signal word ends up all the way to the right.

The UART's error-detection circuits check the incoming data according to the previously selected controls. The *data* bits and *parity* bits are tallied and must be odd or even, as selected on pin 39 (logic 0 is odd, and logic 1 is even). If there is a parity error, the flag flip-flop (to pin 13) is set by going to logic 1. The center of the cell that follows the *parity* bit is tested to see that it is a logic 1. This is the *stop* bit; if it is not present, the framing error flag flip-flop (to pin 14) is set to logic 1.

After the incoming serial data has completely shifted down, a "register-full" condition is detected and the data is transferred in parallel to the output holding register. The "data-available" flag flip-flop, whose output is pin 19, is set, indicating that a character has been received and is ready to be strobed out.

The status bits (parity error on pin 13, data available on pin 19, transmitter input holding register empty on pin 22, framing error on pin 14, and receiver overrun on pin 15) are enabled when pin 16 is at logic 0. Pin 16 is usually hard-wired to ground, except in cases where several UART's are serviced on a common bus. In these cases, each UART is queried by applying a ground pulse to pin 16 of successive UART's. If a data-available condition is detected on pin 19, the processor can read out the data by applying a logic 0 to received data-enable pin 4 of the UART.

Once data is available (as indicated by a logic 1 on pin 19), it must be removed before the next character is shifted all the way into the first register, because new data is written over the old. External circuitry must also clear the data-available flip-flop by applying a logic-0 pulse to reset data available pin 18. If this is not done, the overrun flag flip-flop (pin 15) will be set.



UART MANUFACTURERS

Following is a list of the manufacturers currently making UART's. The numbers in parentheses are the manufacturers' catalog designations.

American Micro-Systems, 3800 Homestead Rd., Santa Clara, CA 95051 (S1757, S1883)
General Instrument, Box 600, Hicksville, NY 11802 (AY-5-1013A)
Intel, 3060 Bowers Ave., Santa Clara, CA 95051 (8201)
Signetics, 811 E. Arques Ave., Sunnyvale, CA 94086 (2536)
SMC Microsystems, 35 Marcus Blvd., Hauppauge, NY 11787 (COM 2017, COM 2502)
Texas Instruments, Box 5012, Dallas, TX 75222 (TMS 6010NC, TMS 6012)
Western Digital, 19242 Red Hill Ave., Newport Beach, CA 92663 (TR1402, TR1602)

In Conclusion. As you can see, the UART could be termed a microprocessor in many respects. Its processing is specialized, dedicated to only the handling of data. Consequently, the UART is "transparent" to the data, since the data is not altered logically in passing through it. What goes into the UART is what comes out.

Considering the number of standard logic gates and register IC's it replaces, you can readily see that the UART is one of the best buys available at its current price of $10 to $15. ◈